

GROVER'S SEARCH WITH LOCAL AND TOTAL DEPOLARIZING CHANNEL ERRORS: COMPLEXITY ANALYSIS

ILAN COHN, ANDRÉ L. FONSECA DE OLIVEIRA and EFRAIN BUKSMAN

Facultad de Ingeniería, Universidad ORT Uruguay, Cuareim 1451

Montevideo, CP 11100, Uruguay

icohn@uni.ort.edu.uy

fonseca@ort.edu.uy

buksman@ort.edu.uy

JESÚS GARCÍA LÓPEZ DE LACALLE

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Universidad Politécnica de Madrid

Calle Alan Turing s/n, 28031 Madrid, Spain

jglopez@etsisi.upm.es

In this article the effect of noise on Grover's algorithm is analyzed, modeled as a total depolarizing channel (TDCh) and a local depolarizing channel in each qubit (LDCh). The focus was not in error correction (e.g. by the fault-tolerant method), but to provide an insight to the kind of error, or degradation, that needs to be corrected. In the last years analytical results regarding mainly the TDCh model have been obtained. In this paper we extend these previous results to the local case, concluding that the degradation of Grover's algorithm with the latter is worse than the former. It has been shown that for both cases with an N -dependent small enough error-width, smaller than $1/\sqrt{N}$ for total error and $1/(\sqrt{N} \log_2 N)$ for the local case, correction is not needed.

I. INTRODUCTION

Quantum systems cannot be completely isolated from the environment. When a quantum system is to be controlled externally, e.g. gate applications, state preparations and others, additional errors are generated. Such effect on quantum algorithms has been studied lately by several authors [1–5].

While there is a fault-tolerant model of quantum computing based on the correction of errors below a certain threshold [6, 7], this method is very expensive in computational resources. Therefore, it is important to diagnose the performance of quantum algorithms affected by noise before making any correction.

Grover's quantum algorithm is an oracle-based search of an unordered database that, without noise, presents a quadratic speedup over the brute-force classical search algorithm [8–10]. However, this advantage is affected drastically if the oracle is faulty [11, 12] or in the presence of noise [13].

Local error models are important due to the fact that any implementation of a quantum circuit are affected by gate errors. In this article we study the effect of a known rate of noise on Grover's search algorithm, extending the work done in [14] with total depolarizing channel error model (TDCh), and also analyzing the effects of local depolarizing channel errors (LDCh). For both model errors, it is shown that quantum speedup disappears when the error rate is constant. Furthermore, we have concluded that the degradation of Grover's algorithm with local error (LDCh) is worse than with total error (TDCh). This is coherent with recent results [15].

The article is divided as follows: in order to introduce the equations used later, in section 2 Grover's algorithm is briefly explained. Sections 3 and 4 deal with the effect of total and local depolarizing error (TDCh and LDCh) on Grover's search, respectively. Some conclusions are drawn in section 5. The major calculus are given in the appendixes.

II. GROVER'S QUANTUM SEARCH ALGORITHM

Grover's quantum search algorithm is known to be optimal, in some sense, to solve the problem of finding a marked element in a unsorted database of N elements using $k_{Gr} = \left\lfloor \frac{\pi}{4} \sqrt{N} \right\rfloor$ oracle queries [16]. Such problem may be shortly stated as follows: suppose we have a database of $N = 2^n$ of quantum states, being n the number of qubits that span the Hilbert space ($\mathcal{H} = \mathbb{C}^{2^n}$), and an unknown marked state among them. Given an *oracle* (or *black box*) to identify whether an element is the one being searched, the goal is to find such marked state with high probability and in as few steps as possible.

Let $|t\rangle$ denote the target basis state and $|s\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$ the superposition of all basis states. To be able to introduce quantum error, we use the density matrix notation ($\mathcal{H} = \mathbb{C}^{2^n} \times \mathbb{C}^{2^n}$), obtaining: $\rho_t = |t\rangle\langle t|$ and $\rho_0 = |s\rangle\langle s|$, respectively. The algorithm is represented in **Algorithm 1**.

Algorithm 1: Grover's search algorithm

1. Set up the superposition state ρ_0 .
 2. Apply the oracle operator $O = 2|t\rangle\langle t| - I$.
 3. Apply the diffusion operator $D = 2|s\rangle\langle s| - I$.
 4. Repeat steps 2 and 3 $\left\lfloor \frac{\pi}{4} \sqrt{N} \right\rfloor - 1$ times.
 5. Perform measurements in the canonical basis in each qubit. The target state will emerge with high probability as $N \gg 1$.
-

It is straightforward that after applying the oracle operator and diffusion operator k times, the result is

$$\rho(k) = G^k \rho_0 (G^\dagger)^k, \quad (1)$$

where $G = DO$ is also known as the Grover operator. It can be shown that the density operator obtained afterwards is

$$\rho(k) = |s_k\rangle\langle s_k|, \quad (2)$$

where

$$|s_k\rangle = \sin((2k+1)\theta)|t\rangle + \cos((2k+1)\theta)|\bar{t}\rangle, \\ |\bar{t}\rangle = \frac{1}{\sqrt{N-1}} \sum_{\substack{i=0 \\ i \neq t}}^{N-1} |i\rangle \text{ and } \theta = \arcsin\left(\frac{1}{\sqrt{N}}\right). \quad (3)$$

Hence, the probability of success (measuring the marked element) after k steps is

$$p(k) = \sin^2((2k+1)\theta). \quad (4)$$

III. GROVER'S SEARCH ALGORITHM WITH TOTAL DEPOLARIZING CHANNEL

In this section we analyze how the error modeled as the Total Depolarizing Channel (TDCh) affects Grover's search algorithm. Similar work, but with different focus, has been done by Vrana et al [14]. Instead of minimizing the mean cost as done in [14] we propose a maximization of the probability, because a closed form expression can be obtained. This enables us to compare the results with the error modeled as the Local Depolarizing Channel (LDCh), in Sec. IV. The TDCh is an error model, which maps an n -qubit state ρ to the maximally mixed state $\frac{I}{N}$ with probability γ , and leaves it unchanged with probability $(1-\gamma)$ [17], given by

$$\varepsilon(\rho, \gamma) = (1-\gamma)\rho + \gamma\frac{I}{N}. \quad (5)$$

This can be interpreted in the probabilistic sense as a Binomial distribution, where *success* or *failure* should be interpreted as having ρ or $\frac{I}{N}$ respectively. We will use these terms indistinctly.

A. State evolution with error

We now consider the effects of the TDCh in Grover's algorithm. That is, every step of the algorithm involves applying the Grover operator, followed by the TDCh error. Since the TDCh error commutes with any unitary operator, the density matrix obtained after k steps is

$$\hat{\rho}(k, \gamma) = (1-\gamma)^k \rho(k) + (1-(1-\gamma)^k) \frac{I}{N}, \quad (6)$$

where $\rho(k)$ is given by Eq. (3). This has a very straightforward interpretation: we have probability of success $(1-\gamma)^k$ and probability of failure $1-(1-\gamma)^k$.

This implies the probability of finding the marked element is

$$\hat{p}(k, \gamma) = (1-\gamma)^k p(k) + \frac{1-(1-\gamma)^k}{N}, \quad (7)$$

where $p(k)$ is given by Eq. (4). Notice how the maximally mixed state $\frac{I}{N}$ introduces uniform probability over all possible states.

As can be seen from Fig. 1 and Fig. 2, the first maxima of the probability $\hat{p}(k)$, at step k_{\max} , moves left with respect to k_{Gr} as either γ or N increase [3]. This can also be appreciated on Fig. 3, which also shows that for any number of qubits, e.g. $n=10$, the behavior is similar. A good approximation of k_{\max} can be deduced by maximizing Eq. (7) (see A), resulting in

$$k_{\max}(\gamma) = \max \left(\left\lfloor \frac{\pi - \arcsin \delta - \arcsin \left(\left[1 - \frac{2}{N}\right] \delta \right)}{4\theta} \right\rfloor, 1 \right) \quad (8)$$

where

$$\delta = \sqrt{\frac{1}{1 + \left(\frac{4\theta}{\ln(1-\gamma)} \right)^2}} \quad (9)$$

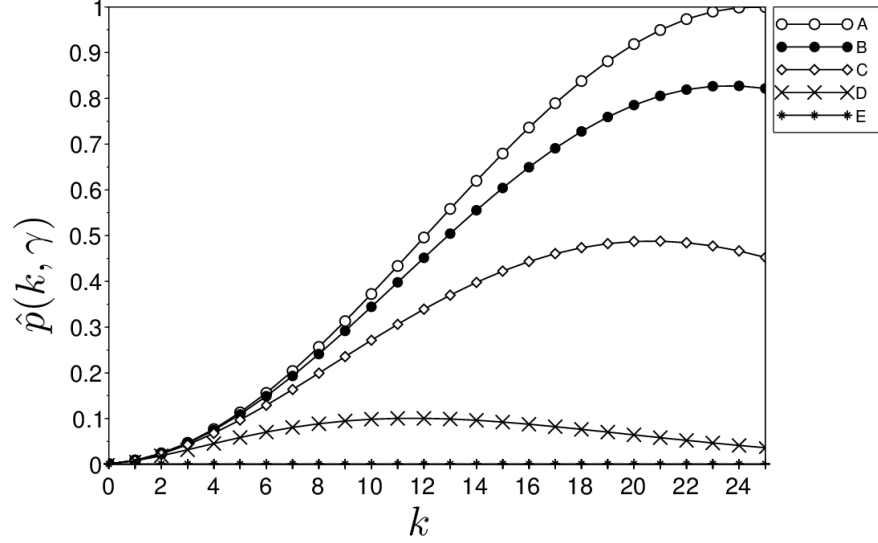


FIG. 1. Probability vs. number of steps with $n = 10$ qubits and different values of γ : 0 (A), $1/(4\sqrt{N})$ (B), $1/\sqrt{N}$ (C), $4/\sqrt{N}$ (D) and 1 (E).

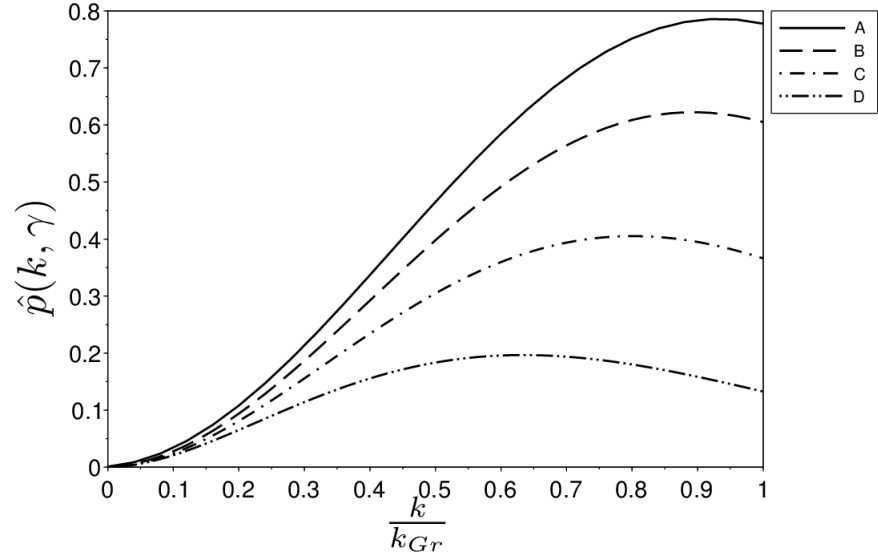


FIG. 2. Probability vs. normalized number of steps with $\gamma = 0.01$ and different number of qubits n : 10 (A), 12 (B), 14 (C) and 16 (D).

and θ is given by Eq. (3). We provide two approximations of Eq. (8): for $n \gg 1$ ($N = 2^n$) and $0 \leq \gamma \ll \frac{2\pi}{\sqrt{N}}$

$$k_{\max}(\gamma) \approx \left\lfloor \frac{\pi\sqrt{N}}{4} - \frac{N\gamma}{8} \right\rfloor, \quad (10)$$

and for $n \gg 1$ with $1 \geq \gamma \gg \frac{8}{\pi\sqrt{N}}$

$$k_{\max}(\gamma) \approx \max \left(\left\lfloor -\frac{1}{\ln(1-\gamma)} + \frac{1}{2} \sqrt{1 + \left(\frac{2}{\ln(1-\gamma)} \right)^2} \right\rfloor, 1 \right). \quad (11)$$

Notice how for Eq. (10), the approximation is valid only for small values of γ ; while the approximation given by Eq. (11) is valid for a large range of γ . Fig. 4 shows different approximations of k_{\max} .

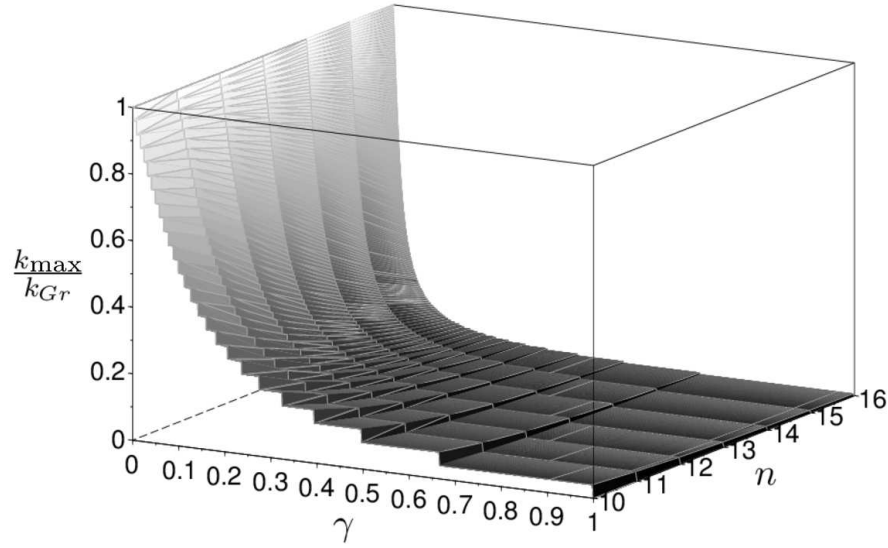


FIG. 3. Number of steps k_{\max}/k_{Gr} vs. width of noise γ vs. number of qubits n .

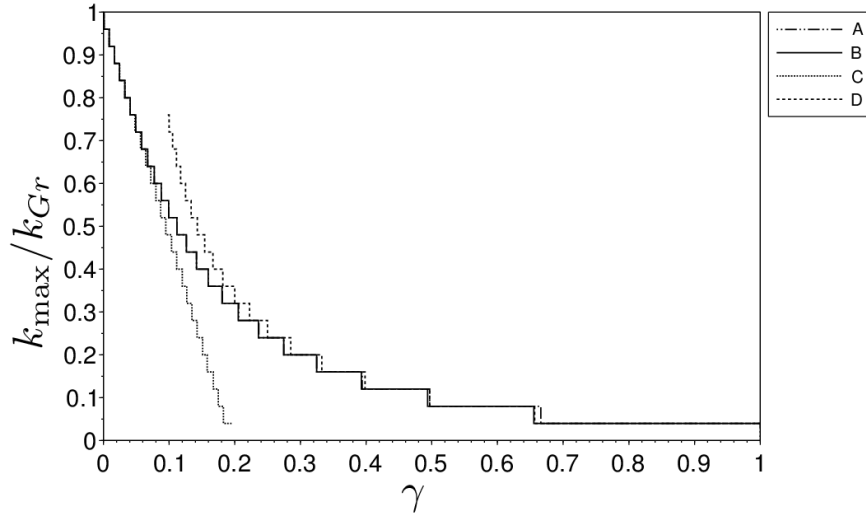


FIG. 4. Step of maximum probability vs. width of noise γ , with $n = 10$ qubits and different approximations of k_{\max} : obtained by simulation (A) and as in equations (8) (B), (10) (C) and (11) (D).

B. Order estimation of the algorithm

If we apply classical amplification and stop at step k , we have the order of the algorithm given by the estimate mean cost (expected number of oracle queries):

$$MC_{k,\gamma} := MC[k, \hat{p}(k, \gamma)] = \frac{k}{\hat{p}(k, \gamma)}. \quad (12)$$

It is well-known that the classical brute-force search algorithm has order $\Theta(N)$ with a constant factor of $1/2$.

1. Stopping the algorithm at k_{Gr}

If we were to stop the algorithm at k_{Gr} , we would have the following estimate cost for the algorithm:

$$MC_{k_{Gr},\gamma} = \frac{k_{Gr}}{(1-\gamma)^{k_{Gr}} p(k_{Gr}) + \frac{1}{N} (1 - (1-\gamma)^{k_{Gr}})}. \quad (13)$$

Since we consider $N \gg 1$, we have: $p(k_{Gr}) \approx 1$ and $k_{Gr} = \left\lfloor \frac{\pi\sqrt{N}}{4} \right\rfloor \approx \frac{\pi\sqrt{N}}{4}$, leaving Eq. (13) as:

$$MC_{k_{Gr}, \gamma} \approx \frac{\frac{\pi\sqrt{N}}{4}}{(1-\gamma)^{\frac{\pi\sqrt{N}}{4}} + \frac{1}{N} \left(1 - (1-\gamma)^{\frac{\pi\sqrt{N}}{4}}\right)}. \quad (14)$$

Notice that for any fixed $\gamma > 0$ the order is $\Theta(N\sqrt{N})$. For small values of γ ($\frac{\pi\sqrt{N}}{4}\gamma \ll 1$) Eq. (13) becomes

$$MC_{k_{Gr}, \gamma} \approx \frac{\pi\sqrt{N}}{4} \left(1 + \frac{\pi\sqrt{N}}{4}\gamma\right). \quad (15)$$

It is interesting to find the width of noise that keeps the quantum speedup (order of the algorithm $\Theta(\sqrt{N})$). Hence, it is straightforward that γ needs to be a function of N . In fact, one needs $\gamma \ll 1/k_{Gr}$, as shown in [14].

However, Grover's quantum search algorithm can still outperform the brute-force classical search algorithm. In fact, there is a maximum width of noise ($\gamma_{\text{classical}}$) that leaves the mean cost of Grover's quantum search algorithm $\frac{N}{2}$. Such is found by using Eq. (14):

$$\gamma_{\text{classical}} \approx 1 - \left(\frac{-1 + 2k_{Gr}}{-1 + N}\right)^{\frac{1}{k_{Gr}}}, \quad (16)$$

which for $N \gg 1$, becomes

$$\gamma_{\text{classical}} \approx \frac{4 \ln \left(\frac{2\sqrt{N}}{\pi}\right)}{\pi\sqrt{N}}. \quad (17)$$

2. Stopping the algorithm at k_{\max}

If we were to stop the algorithm at $k_{\max}(\gamma)$, Eq. (12) becomes $MC[k_{\max}(\gamma), \hat{p}(k_{\max}(\gamma), \gamma)]$. Therefore, in contrast to stopping the algorithm at k_{Gr} , for any fixed $\gamma > 0$ the algorithm is of order $\Theta(N)$. For large values of N ($N \gg 1$) and small values of γ ($\sqrt{N}\gamma/(2\pi) \ll 1$), the mean cost becomes

$$MC_{k_{\max}(\gamma), \gamma} \approx \frac{\pi\sqrt{N}}{4} \left(1 + \frac{\pi\sqrt{N}}{4}\gamma \left(1 - \frac{2}{\pi^2}\right)\right). \quad (18)$$

Comparing this result with Eq. (15), we can see that an extra constant factor of approximately 0.80 appears. Then, we need $\gamma \ll 1/(0.8k_{Gr})$ to maintain the quadratic speedup (order of the algorithm $\Theta(\sqrt{N})$).

For any fixed $\gamma > 0$, the mean cost becomes

$$MC_{k_{\max}(\gamma), \gamma} \approx \frac{N}{9 - 8\gamma}. \quad (19)$$

As a result, the maximum width of noise that leaves $MC[k_{\max}(\gamma), \hat{p}(k_{\max}(\gamma), \gamma)]$ classical ($N/2$) is independent of N , i.e. $\gamma_{\text{classical}} \approx 7/8$.

IV. GROVER'S SEARCH ALGORITHM WITH LOCAL DEPOLARIZING CHANNEL

The Local Depolarizing Channel (LDCh), maps an n -qubit state ρ to a mixed state applying the depolarizing channel in every qubit independently, i.e.

$$\varepsilon(\rho) = \varepsilon_1(\rho, \alpha) \circ \varepsilon_2(\rho, \alpha) \circ \dots \circ \varepsilon_n(\rho, \alpha), \quad (20)$$

where $\varepsilon_i(\rho, \alpha)$ corresponds to the depolarizing channel (Eq. (5)) acting in the qubit i , and α the probability of error. Analogously to Sec. III, we define the probability of obtaining the marked element after k steps as $\hat{p}^L(k, \alpha)$, and the step at which the first maxima of the probability occurs as $k_{\max}^L(\alpha)$.

A. Operator-sum representation for Grover state evolution with LDCh

For one qubit states the Depolarizing Channel error (Eq. (5)) can be expressed using Kraus' operators (Operator-sum representation), as

$$\varepsilon(\rho, \alpha) = \sum_{l=0}^4 M_l \rho M_l^\dagger, \quad (21)$$

where $M_0 = \sqrt{(1 - \frac{3\alpha}{4})} \sigma_0$ and $M_i = \sqrt{\frac{\alpha}{4}} \sigma_i$ ($i \in \{1, 2, 3\}$), being σ_μ the Pauli matrices.

In the case applying local errors in all qubits in a n -qubits state there are 4^n operators (all possible combinations), which are of the form [17]

$$M_c = \sqrt{\left(1 - \frac{3\alpha}{4}\right)^m \left(\frac{\alpha}{4}\right)^{n-m}} \bigotimes_{q=1}^n \sigma_{i_q}, \quad (22)$$

where σ_{i_q} is the i_q Pauli matrix ($i_q \in \{0, \dots, 3\}$) and m is the amount of σ_0 (I) operators present in a particular M_c .

If we consider one step of the algorithm as applying the Grover operator G and the LDCh, we get the following state after k steps

$$\hat{\rho}^L(k, \alpha) = \sum_{c_1}^{4^n} \cdots \sum_{c_k}^{4^n} \left[O_\Pi \rho_{\text{ini}} O_\Pi^\dagger \right], \quad (23)$$

where

$$O_\Pi = \prod_{l=1}^k (M_{c_l} G), \quad (24)$$

l represents the step, each M_{c_l} has the form of Eq. (22) and $\rho_{\text{ini}} = |s\rangle\langle s|$ is the initial Grover state.

For the first step ($k = 1$), such probability is

$$\begin{aligned} \hat{p}^L(1, \alpha) &= \frac{1}{2^{3n-4}} \left[2^n (2^{n-1} - 1) \left(1 - \frac{\alpha}{2}\right)^n + \right. \\ &\quad \left. + \frac{1}{16} (2^n - 4)^2 \right], \end{aligned} \quad (25)$$

which for $n \gg 1$ becomes

$$\hat{p}^L(1, \alpha) \approx \frac{1}{2^n} \left[2^3 \left(1 - \frac{\alpha}{2}\right)^n + 1 \right]. \quad (26)$$

Due to the exponentially increasing complexity of Eq. (23), we propose to analyze $\hat{p}^L(k, \alpha)$ using lower and upper bounds in terms of the TDCh model (because of simplicity and relatively tight bounds), and a first order approximation.

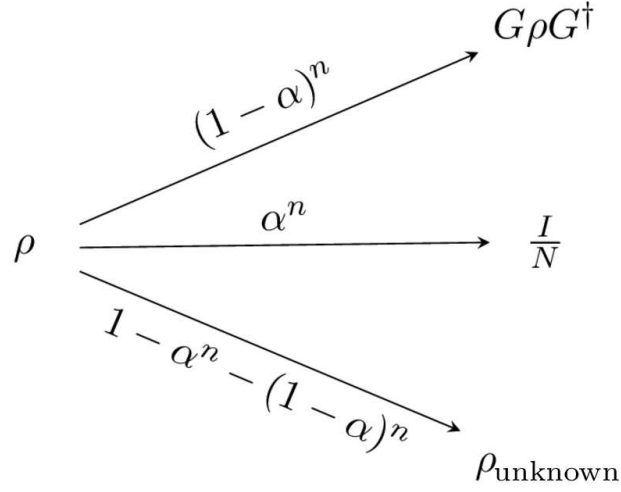
B. Probability Bounds

Grover's algorithm with LDCh error can be interpreted as a ternary tree where each depth-level corresponds to a step. The initial node is given by $\rho_{\text{ini}} = |s\rangle\langle s|$, and in each node we have the paths shown in Fig. 5. In fact, the actual probability of getting the marked element at a step k is obtained by taking all the possible paths (which are 4^{nk} if one uses the Kraus' operators).

In order to find upper and lower probability bounds, we assume the best and worst case for the desired probability, by taking $\rho_{\text{unknown}} \equiv G\rho G^\dagger$ or $\rho_{\text{unknown}} \equiv I/N$, respectively. Thus, the probability bounds are

$$\hat{p}_u(k, \alpha) = (1 - \alpha^n)^k p(k) + \frac{1 - (1 - \alpha^n)^k}{N} \quad (27)$$

$$\hat{p}_l(k, \alpha) = (1 - \alpha)^{nk} p(k) + \frac{1 - (1 - \alpha)^{nk}}{N}, \quad (28)$$

FIG. 5. Ternary tree at step k of the algorithm with LDCh.

that hold

$$\hat{p}_l(k, \alpha) \leq \hat{p}^L(k, \alpha) \leq \hat{p}_u(k, \alpha), \quad (29)$$

$\forall k \in \mathbb{N}, \alpha \in [0, 1]$. Since these are of the form of the TDCh (Eq. (6)), we define these bounds as their equivalent width of error $\gamma(\alpha)$, i.e.

$$\gamma_u(\alpha) = \alpha^n, \quad (30)$$

$$\gamma_l(\alpha) = 1 - (1 - \alpha)^n. \quad (31)$$

Even though $\gamma_l(\alpha)$ provides a reasonable probability bound, $\gamma_u(\alpha)$ does not (see Fig. 6). A better approach is to replace it by

$$\gamma_u(\alpha) = \frac{n\alpha}{2 + n\alpha}. \quad (32)$$

C. First order approximation of the LDCh

Here we will propose a method for analyzing the effect of the LDCh in Grover's algorithm in terms of its first order approximation (in α , as in [17]), proving also that this approximation serves both as an approximation for small values of width of error α and as a lower probability bound for any α .

After applying the LDCh in every qubit and every step (Eq. (23)), the probability of success after k steps is of the form

$$\hat{p}^L(k, \alpha) = \sum_{i=0}^{nk} \left(1 - \frac{3\alpha}{4}\right)^{nk-i} \left(\frac{\alpha}{4}\right)^i f_i(n, k), \quad (33)$$

where i is the total number of *errors* up to step k . Hence, the first order approximation is

$$\hat{p}^L(k, \alpha) \approx \left(1 - \frac{3\alpha}{4}\right)^{nk} f_0(n, k) + \left(1 - \frac{3\alpha}{4}\right)^{nk-1} \left(\frac{\alpha}{4}\right) f_1(n, k), \quad (34)$$

where $f_0(n, k) \equiv p(k)$ is Grover's algorithm original probability and $f_1(n, k)$ corresponds to the all the contributions of applying *one* error in any qubit at any step of the algorithm (see B). As can be seen in Fig. 6, Eq. (34) is also a lower bound of the probability because all $f_i(n, k)$ are non-negative (sums of probabilities).

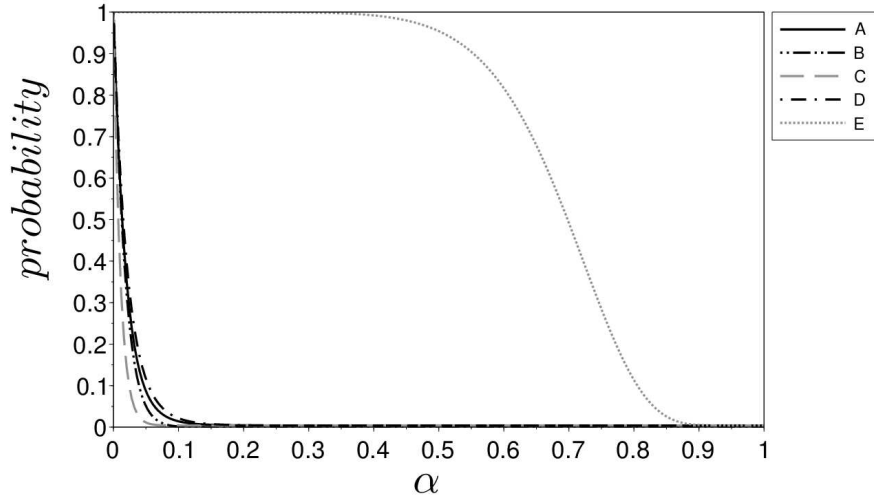
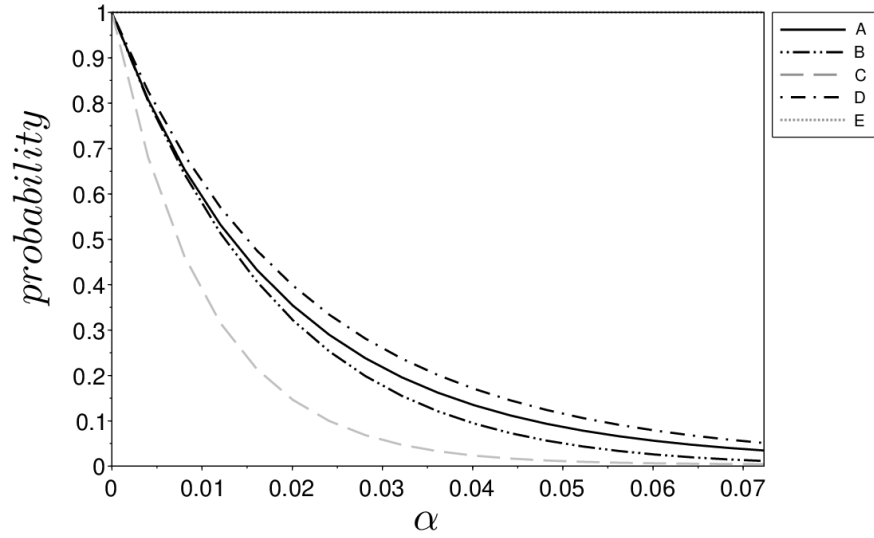
(a) Probability evaluated at k_{Gr} vs. width of error α for $n = 8$ qubits and its bounds.(b) Detail for small values of α .

FIG. 6. Comparison between different bounds of probabilities: numerical simulation (A), first order approximation (B), lower bound with $\gamma_l(\alpha) = 1 - (1 - \alpha)^n$ (C), upper bound with $\gamma_u(\alpha) = n\alpha/(2 + n\alpha)$ (D) and upper bound with $\gamma_u(\alpha) = \alpha^n$ (E).

D. Order estimation of the algorithm

Here we denote the estimate mean cost of the algorithm with LDCh error by $MC_{k,\alpha}^L := MC^L[k, \hat{p}^L(k, \alpha)]$. Using the monotonicity of both $\hat{p}^L(k, \alpha)$ and $(\gamma_l - \gamma_u)$ in α , and Eq. (29), we have found bounds for the mean cost as follows:

$$MC_{k,\gamma_u} \leq MC_{k,\alpha}^L \leq MC_{k,\gamma_l}. \quad (35)$$

That is, given a fixed number of steps k , the bounds $\hat{p}(k, \gamma_u)$ and $\hat{p}(k, \gamma_l)$ induce bounds on the mean cost.

1. Stopping the algorithm at k_{Gr}

Considering k_{Gr} as the final step, we have an estimate mean cost of the algorithm given by $MC_{k_{Gr},\alpha}^L$, bounded as in Eq. (35). Similar to the TDCh case, for any fixed $\alpha > 0$ the order is $\Theta(N\sqrt{N})$.

For $N \gg 1$ and small values of α ($\alpha k_{Gr} \log_2 N \ll 1$) the mean cost becomes

$$MC_{k_{Gr}, \alpha}^L = \frac{\pi\sqrt{N}}{4} + \zeta \alpha N \pi^2 \log_2 N, \quad (36)$$

where ζ is a constant in the interval $[1/32, 1/16]$ (from bounds). Using the first order approximation, Eq. (34) and $f_1(n, k)$, we have found that ζ is approximately 70/2048. The latter is much closer to $1/32$ than to $1/16$, and hence, the actual probability is much closer when using γ_u than when using γ_l . Because of the simplicity and the validity of the expression, we encourage to use γ_u to estimate the probability $\hat{p}^L(k, \alpha)$.

From Eq. (36), one can see that to maintain the quadratic speedup $\alpha \ll 2/(k_{Gr} \log_2 N)$ is needed.

2. Stopping the algorithm at k_{\max}

We now consider the induced bounds for k_{\max} , which are justified by the continuity of $\hat{p}(k, \gamma)$ in both γ and n . These are

$$k_{\max, l} \leq k_{\max}^L \leq k_{\max, u}, \quad (37)$$

where $k_{\max, l} = k_{\max}(\gamma_l)$ and $k_{\max, u} = k_{\max}(\gamma_u)$. Since we are stopping at the step of maximum probability (in each curve), the following bound holds:

$$MC_{k_{\max, l}, \gamma_l} \leq MC_{k_{\max}^L, \alpha}^L \leq MC_{k_{\max, u}, \gamma_u}. \quad (38)$$

In fact, using Eqs. (38) and (26), one finds that $k_{\max}^L \rightarrow 1$ and the mean cost is given by $1/p^L(1, \alpha)$. Therefore, again, for any fixed $\alpha > 0$ the order is $\Theta(N)$.

For $N \gg 1$ and small values of α ($0.55\alpha N \log_2 N \ll 1$), the mean cost is approximately

$$MC_{k_{\max}^L, \alpha}^L = \frac{\pi\sqrt{N}}{4} + \zeta \alpha N \log_2 N, \quad (39)$$

where ζ is a constant in the interval $[(\pi^2 - 4)/32, (\pi^2 - 1)/16]$. And for the algorithm to be of order $\Theta(\sqrt{N})$, one needs $\alpha \ll 2.5/(k_{Gr} \log_2 N)$.

This is coherent with the results presented with the TDCh model, where for small values of γ , choosing k_{Gr} or k_{\max} does not change the order of the algorithm, but for any fixed $\gamma > 0$, it changes drastically.

Analogously to the work done in Sec. III, one can find $\alpha_{\text{classical}}$: the maximum value of α that leaves the mean cost $N/2$. Considering the worst case yields

$$\alpha_{\text{classical}} \leq \frac{14}{\log_2 N}. \quad (40)$$

In contrast to the TDCh case, this width of error does depend on N .

V. CONCLUSIONS

In this article we have studied the effect of noise in Grover's quantum search algorithm, based in two models of error: total depolarizing channel (TDCh) and local depolarizing channel (LDCh). Our focus was not in error correction (e.g. by the fault-tolerant method), but to provide an insight to the kind of error, or degradation, that needs to be corrected. Our work extends the results found by Vrana et. al. regarding only the TDCh model, concluding that the LDCh error model is more devastating than the TDCh error. We also show that, for both cases, with an N -dependent small enough error-width, correction is not needed.

The devastating effect of the LDCh was, in some sense, expected: in the case of TDCh model the state is partially altered, but the rest remains intact (the state evolves partially in the plane formed by $|s\rangle$ and $|t\rangle$, or otherwise becomes I/N). Whereas, with an LDCh error model, the state is mostly removed from the plane formed by $|s\rangle$ and $|t\rangle$.

We have found a closed form for the step at which the maximum of probability occurs for the TDCh error model (k_{\max}), and provided bounds for the LDCh error model (based on the TDCh model). This step could be used as an indicator of the impact of the error in the algorithm. This means, comparing k_{\max} with k_{Gr} one has an idea of how degraded the algorithm is. Analogous comparisons could be done with the probability at such steps, but are, in some sense, equivalent.

From this study we conclude that to maintain the original speed up of Grover's algorithm, different requirements are needed depending on the error model: in the TDCh case, the error-width has to be smaller than $1/\sqrt{N}$ (as in [14]); and for the LDCh model, it needs to be smaller than $1/(\sqrt{N} \log_2 N)$.

Furthermore, there are widths of error for both the TDCh and the LDCh model for which the algorithm still outperforms the classical brute-force algorithm. It is interesting to note that for the TDCh model this width does not depend on N , i.e. for widths below $7/8$ the algorithm is better than the classical (Eq. (19)). Nevertheless, for the LDCh model, this classical threshold error $\alpha_{\text{classical}}$, depends of N (Eq. (40)) and goes to zero with N , reinforcing the idea that the LDCh degrades more the algorithm than the TDCh.

The depolarizing channel is a particular error model. It would be interesting to study other error models taking into account experimental implementations of the algorithm. Future work should be done with this focus, and possibly including error correction.

Appendix A: Finding k_{\max}

We wish to find k_{\max} which maximizes Eq. (7). We will assume $k \in \mathbb{R}$, and then perform some adjustments. Considering $p(k)$ as in Eq. (4), we find the first derivative of Eq. (7)

$$\frac{d\hat{p}(k, \gamma)}{dk} = (1 - \gamma)^k \left[2\theta \sin(2(2k + 1)\theta) + \left(p(k) - \frac{1}{N} \right) \ln(1 - \gamma) \right]. \quad (\text{A1})$$

Assuming $\gamma \neq 1$ and nulling the derivative yields

$$4\theta \sin((2k + 1)\theta) \cos((2k + 1)\theta) + \left(\sin^2((2k + 1)\theta) - \frac{1}{N} \right) \ln(1 - \gamma) = 0. \quad (\text{A2})$$

The former can be expressed as

$$-C = -A \sin^2 x + B \sin x \cos x, \quad (\text{A3})$$

where $x = (2k + 1)\theta$, $A = -\ln(1 - \gamma)$, $B = 4\theta$ and $C = -\ln(1 - \gamma)/N$. This can be reduced to

$$-z = \sin(x) \cos(x + \phi), \quad (\text{A4})$$

where $z = C/\sqrt{A^2 + B^2}$ and $\phi = \arcsin(A/\sqrt{A^2 + B^2})$. Considering $x > 0$ Eq. (A4) has two family of solutions

$$\begin{aligned} x_1 &= \pi m - \frac{1}{2} \arcsin(2z - \sin(\phi)) - \frac{\phi}{2}, \quad \text{and} \\ x_2 &= \frac{1}{2} (2\pi m + \arcsin(2z - \sin(\phi)) - \phi + \pi), \end{aligned} \quad (\text{A5})$$

being m an arbitrary integer. Thus, the solution to the original problem is

$$\frac{\pi - 2\theta - \arcsin(\delta) - \arcsin\left(\left[1 - \frac{2}{N}\right]\delta\right)}{4\theta}. \quad (\text{A6})$$

where

$$\delta = \sqrt{\frac{1}{1 + \left(\frac{4\theta}{\ln(1-\gamma)}\right)^2}}. \quad (\text{A7})$$

Adjusting border values with $\gamma \rightarrow 0$ and $\gamma \rightarrow 1$, and taking integer part yields

$$k_{\max}(\gamma) = \max \left(\left\lfloor \frac{\pi - \arcsin \delta - \arcsin\left(\left[1 - \frac{2}{N}\right]\delta\right)}{4\theta} \right\rfloor, 1 \right). \quad (\text{A8})$$

Appendix B: Finding $f_1(n, k)$

As stated before, $f_1(n, k)$ corresponds to the depolarizing channel acting in any qubit at any step of the algorithm. Because of the symmetry of Grover's algorithm in the qubits, one can assume without loss of generality that the last qubit is the affected one and then multiply by the number of possible ways of having the depolarizing channel acting on one qubit (n). We can also assume, without loss of generality, that the target state is $|t\rangle \equiv |00 \cdots 0\rangle = |0\rangle$.

The depolarizing channel acting on one qubit (without the identity operator and considering that error has been committed) can be expressed with the operators

$$X := I^{\otimes n-1} \otimes \sigma_x, \quad Y := I^{\otimes n-1} \otimes \sigma_y, \quad Z := I^{\otimes n-1} \otimes \sigma_z, \quad (\text{B1})$$

where I represents the 1-qubit identity operator and σ_μ the Pauli matrices.

It is useful to see that the set $\{|0\rangle, |s\rangle, |1\rangle, |p\rangle\}$ is closed under the operations of $\{X, Y, Z, G\}$, where G is Grover's operator and $|p\rangle$ is defined as the normalized superposition of the even states of the canonical base as

$$|p\rangle := \sqrt{\frac{2}{N}} \sum_{i=0}^{\frac{N}{2}-1} |2i\rangle. \quad (\text{B2})$$

The state obtained after k steps is represented as $\phi(k) = [a_k \ b_k \ c_k \ d_k]^T$, which corresponds to the quantum state $|\varphi_k\rangle = a_k|0\rangle + b_k|s\rangle + c_k|1\rangle + d_k|p\rangle$. The probability of obtaining the marked state ($|0\rangle$) is given by $[\text{tr}(C \cdot \phi(k))]^2$, where $C = \begin{bmatrix} 1 & \lambda/2 & 0 & \lambda/\sqrt{2} \end{bmatrix}$ and $\lambda = 2/\sqrt{N}$. Hence, the desired function $f_1(n, k)$ is

$$f_1(n, k) = \sum_{l=1}^k \sum_{j=1}^3 [\text{tr}(C A^{k-l} B_j A^l \phi(0))]^2, \quad (\text{B3})$$

where $\phi(k) = [0 \ 1 \ 0 \ 0]^T$ is the initial step of the algorithm ($|s\rangle$) and A, B_1, B_2, B_3 correspond to applying G, X, Y , or Z respectively (up to a global phase). Such matrices are given by

$$A = \begin{bmatrix} -1 & -\lambda & 0 & -\sqrt{2}\lambda \\ \lambda & \lambda^2 - 1 & -\lambda & \sqrt{2}(\lambda^2 - 1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & \sqrt{2} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad (\text{B4})$$

$$B_2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & \sqrt{2} \\ 1 & 0 & 0 & 0 \\ 0 & -\sqrt{2} & 0 & -1 \end{bmatrix} \quad \text{and} \quad B_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -\sqrt{2} & 0 & 1 \end{bmatrix}. \quad (\text{B5})$$

The expression of $f_1(n, k)$, derived here, has been used to perform the series expansion made in Eq. (36). The full expression is too large to include in this article.

-
- [1] Hiroo Azuma. Decoherence in grover's quantum algorithm: Perturbative approach. *Physical Review A*, 65:042311, Apr 2002.
 - [2] Daniel Shapira, Shay Mozes, and Ofer Biham. Effect of unitary noise on grover's quantum search algorithm. *Physical Review A*, 67:042301, Apr 2003.
 - [3] P. J. Salas. Noise effect on grover algorithm. *The European Physical Journal D*, 46(2):365–373, 2008.
 - [4] Piotr Gawron, Jerzy Klamka, and Ryszard Winiarczyk. Noise effects in the quantum search algorithm from the viewpoint of computational complexity. *Applied Mathematics and Computer Science*, 22(2):493–499, 2012.
 - [5] Andris Ambainis, Arturs Backurs, Nikolajs Nahimovs, and Alexander Rivosh. Grover's algorithm with errors. In Antonin Kucera, Thomas A. Henzinger, Jaroslav Nesetril, Tomas Vojnar, and David Antos, editors, *Mathematical and Engineering Methods in Computer Science*, volume 7721 of *Lecture Notes in Computer Science*, pages 180–189. Springer Berlin Heidelberg, 2013.
 - [6] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 176–188, New York, NY, USA, 1997. ACM.

- [7] Daniel Gottesman. Theory of fault-tolerant quantum computation. *Physical Review A*, 57:127–137, Jan 1998.
- [8] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings, 28th Annual ACM Symposium on the Theory of Computing* p. 212, 1996.
- [9] L. K. Grover. Quantum mechanics helps in searching for a needle in haystack. *Phys. Rev. Lett.*, 79(2):325–328, July 1997.
- [10] Lov K. Grover. From schrdingers equation to the quantum search algorithm. *American Journal of Physics*, 69(7):769–777, 2001.
- [11] Oded Regev and Liron Schiff. Impossibility of a quantum speed-up with a faulty oracle. In Luca Aceto, Ivan Damgrd, LeslieAnn Goldberg, MagnsM. Halldrsson, Anna Inglsdttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, volume 5125 of *Lecture Notes in Computer Science*, pages 773–781. Springer Berlin Heidelberg, 2008.
- [12] Kristan Temme. Runtime of unstructured search with a faulty hamiltonian oracle. *Physical Review A*, 90:022310, Aug 2014.
- [13] Jingling Chen, Dagomir Kaszlikowski, L.C. Kwek, and C.H. Oh. Searching a database under decoherence. *Physics Letters A*, 306(56):296 – 305, 2003.
- [14] Peter Vrana, David Reeb, Daniel Reitzner, and Michael M. Wolf. Fault-ignorant quantum search. *New Journal of Physics*, 16:073033, 2014.
- [15] Rafał Demkowicz-Dobrzański and Marcin Markiewicz. Quantum computation speedup limits from quantum metrological precision bounds. *Physical Review A*, 91:062322, Jun 2015.
- [16] Christof Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, Oct 1999.
- [17] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge Univ. Press, 2000.